1-1-2000

Securing wavelet compression with random permutations

Takeyuki Uehara
University of Wollongong, takeyuki@uow.edu.au

Reihaneh Safavi-Naini
University of Wollongong, rei@uow.edu.au

Philip Ogunbona
Motorola Australian Research Centre, philipo@uow.edu.au

Securing wavelet compression with random permutations

Abstract

Wavelet compression for digital images achieves very high compression with reasonably high image quality and so is widely used in various applications. Adding security to compression algorithms has been proposed in a number of compression systems with the aim reducing the overall cost of compression and encryption. In this paper we propose a combined compression and encryption system based on wavelet transform and examine its security. Our results show that with a relatively small added cost varying degrees of security can be obtained while maintaining the performance of the compression system.

Keywords

random, compression, wavelet, permutations, securing

Disciplines

Physical Sciences and Mathematics

Publication Details

Uehara, T., Safavi-Naini, R. & Ogunbona, P. (2000). Securing Wavelet Compression with Random Permutations. The First IEEE Pacific-Rim Conference on Multimedia (pp. 332-335). Piscataway, NJ, USA: IEEE - Institute of Electrical and Electronics Engineers Inc..

Securing Wavelet Compression with Random Permutations

Takeyuki Uehara [1], Reihaneh Safavi-Naini [1], Philip Ogunbona [2]
[1] School of Information Technology and Computer Science, University of Wollongong
Northfields Ave, Wollongong, NSW 2522, Australia, {tu01,rei}@uow.edu.au
[2] Visual Information Processing Lab, Motorola Labs, Motorola Australian Research Centre
Level 3, 12 Lord Street, Botany NSW 2019, Australia, philip.ogunbona@motorola.com

Abstract

Wavelet compression for digital images achieves very high compression with reasonably high image quality and so is widely used in various applications. Adding security to compression algorithms has been proposed in a number of compression systems with the aim reducing the overall cost of compression and encryption. In this paper we propose a combined compression and encryption system based on wavelet transform and examine its security. Our results show that with a relatively small added cost varying degrees of security can be obtained while maintaining the performance of the compression system.

1 Introduction

Wavelet compression for digital images achieves very high compression with reasonably high image quality. Adding security to compression algorithms has been an attractive proposition especially in the context of electronic commerce applications which require various levels of security and have to cater for other system constraints such as low bandwidth and/or storage. Traditional methods of encrypting the compressed data, although provide very high security but could be unacceptable for many applications due to the added computation and hardware and/or software costs. So flexible systems that can provide various levels of security at varying implementation costs are needed.

To add security to a compression system a key dependent transformation must be used. A *key* is a secret piece of information that is used during the encryption and allows its owner to correctly recover the message while preventing everyone else from accessing the message. We use a simple key dependent transformation, that is a family of permutations indexed by the key, to transform the wavelet coefficients. The decoder applies the inverse permutation before processing the coefficients while the unauthorized user who does not know the key is prevented from recovering the image. The amount of masking will depend on the number of subbands to which the permutation is applied.

In this paper we investigate security and efficiency of the above system using a specific implementation [Dav97]. We will show that this basic system can provide various degrees of masking of information and does not have drastic effect on the compression ratio. To assess security of the system we will examine possible attacks. In particular, we demonstrate a plaintext attack that uses a number of well chosen transform coefficients to derive the secret permutation. We argue that if higher security is required a block cipher algorithm can be used to mask the coefficients of the lowest subband while permutations are used for the other subbands.

The paper is organized as follows. First we briefly review the wavelet compression system used in our experiments and then in Section 3.1 present the encryption scheme using a random permutation. In Section 4 we present our attack and using block encryption to enhance security, and in Section 6 conclude the paper.

2 Wavelet Image Compression

In a wavelet compression system an image is decomposed into *subbands* which are represented by real-valued wavelet coefficient sets. The transform stage is followed by a *quantization* stage which converts the real-valued coefficients to whole numbers. Finally an *entropy coder* is used to compress the output of the quantizer. The transform stage is invertible and the compression is the result of quantization and entropy coding stages. There are numerous approaches to quantization with varying levels of performance.

2.1 Transform

Discrete wavelet transform consists of a sequence of wavelet filter banks. In the 2-dimensional transform, firstly each row of the image is decomposed into coarse and detailed parts and is down-sampled, and then each column of the coarse and detailed parts is decomposed into coarse and detailed parts and is down-sampled again. This results in four parts: coarse-

This research is partly supported by Motorola Australian Research Centre.

coarse, coarse-detail (both from the row coarse part), detail-coarse and detail-detail parts (both from row detailed part). The last three parts compose the output subbands while the coarse-coarse part is the input of the next filter bank. Hence each filter bank produces three subbands and four subbands for the last filter bank.

3 Encryption Using Random Permutation

Using key dependent permutations on transform coefficients of Discrete Cosine Transform (DCT) has been proposed for adding encryption to MPEG coded data [Tan96] [SSR99]. In using a permutation to wavelet transform coefficients the following points must be taken into account. Firstly, DCT transform is used on fixed size ($n \times n$) blocks of pixels in the image and produce the same number of coefficients for each block, while wavelet coefficients are computed for the whole image and so the number of coefficients depends on the image size. Secondly, the quantization precision will be subband dependent and so the number of bits allocated to each coefficient will vary for different subbands. This means that the permutation must be applied to each subband separately, otherwise high compression drop or distortion can be expected.

We use a subband-based permutation system with a different permutation used for each subband. Let $V^{(b)} = (v_0^{(b)}, v_1^{(b)}, v_2^{(b)}, \ldots, v_i^{(b)}, \ldots)$ denote the wavelet coefficient block in subband b, where $v_i^{(b)}$ denotes the ith coefficient in the subband and $|V^{(b)}|$ is the number of coefficients. The permutation can be one that permutes all $|V^{(b)}|$ coefficients, or we may break the block into sub-blocks and permute the coefficients in each sub-block. For simplicity we assume the former.

There are $|V^{(b)}|!$ permutations for subband b and so the number depends on the image size and the order (b) of the subband. Let $K^{(b)}$ be the key used for generation of a permutation for subband b. Keys can be chosen independently for each subband, or by using a master key $\hat{K}$ and a key generation algorithm that has $\hat{K}$ as input and produces keys for all subbands.

Permutation of coefficients may be implemented in the following two ways. *1)* Wavelet coefficients in each subband are permuted after transformation and before quantization. *2)* The quantized coefficients in each subband are permuted after quantization and before entropy coding.

The implementation uses an arithmetic coder to encode the quantized coefficients. The coder encodes the i^{th} bits of all the coefficients in a subband, starting from the most significant bit position and moving to the least significant bit position. Thus coefficients are not separable and so the coefficient-wise permutation is not possible. The two methods could result in different compression ratio. If the quantization method depends on the order of coefficients (context), for example vector quantization, the compression ratio in case **1** will be affected. In both cases if the entropy coding is sensitive to the order of quantized coefficients a drop in compression ratio will be expected.

3.1 Experiments

Permuted subband # in encoding	I-permuted subband # in decoding	Comp. ratio (bpp)	PSNR
None	None	0.9975	39.448
0	None	0.9975	13.051
1	None	0.9975	20.947
2	None	0.9975	28.620
3	None	0.9975	26.837
4	None	0.9975	24.494
5	None	0.9975	29.814
6	None	0.9975	29.900
7	None	0.9983	26.995
8	None	0.9978	31.359
9	None	0.9978	31.722
10	None	1.0014	29.946
11	None	0.9993	33.460
12	None	0.9995	34.907
13	None	1.0040	34.003
14	None	1.0005	36.746
15	None	0.9982	38.602
0 to 7	None	0.9983	11.634
8 to 15	None	1.0162	24.979
0 to 15	None	1.0168	11.444
0 to 15	0 to 15	1.0168	39.448

Table 1: Compression ratio and PSNR with permuted subbands when the target compression ratio is specified to 8:1

We used an implementation [Dav97] that uses Antonini wavelet [ABMD92], employs 2-dimensional transform, has five filter banks and decomposes the image into $3 \times 5 + 1 = 16$ subbands. The program takes the compression ratio as an input parameter and adjusts the quantization precisions of subbands to achieve the compression ratio. The exact compression ratio may not be achievable simply because 1 bit change in the precision of subband b results in $|V^{(b)}|$ bits change in the quantizer output size and the output size will be decreased by $|V^{(b)}|$ bits. If the permutation results in considerable drop in the compression the precisions in some of the subbands will be reduced to achieve the given compression ratio and will result in lower quality image and drop in PSNR (Peak Signal to Noise Ratio).

The test image is `lena.pgm`, which is a 512×512 image with 256 level gray scale (8 bits/pixel). The compression ratio was set to 8:1, ie. 1 bit/pixel (bpp).

Results of the experiment are shown in Table 1 and Figure 1 - 3. Firstly the image is encoded and decoded without permutations and PSNR is calculated. Then coefficients in various subbands are permuted and PSNR is calculated. It can be seen that the permutation of coarser subbands results in a larger PSNR drop compared to the detailed part. In all the above cases, the drop in compression ratio is less than 2%. This is mainly due to the type of the quantization and entropy coding algorithms: the quantization algorithm processes one coefficient at a time and does not depend on other coefficients in the subband, and the entropy coder uses context information which is orthogonal to the direction of the permutation. The number of permuted subbands does not change the precisions of the quantization process because the drop in the compression ratio is not large enough to change the precisions. In the experiment the permutations were applied between the transform and the quantization but similar results can be expected if the permutations are used after the quantization because of the above reasons. The permutation has a small impact on the compression ratio in the implementation.

4 Chosen Plaintext Attack

To evaluate security of the system, the first step is to find the number of keys which in the above system can be chosen to be very large. This ensures that an *exhaustive search attack* will be infeasible. However this would not guarantee security of the system as more efficient attacks could be possible. In the following we describe a chosen plaintext attack that recovers the secret permutation by examining the system output on a number of well chosen attack images.

In a *chosen plaintext attack*, the attacker has access to an encoder with a secret key. He can choose a plaintext, ie. an image, obtain the corresponding ciphertext and analyze the relationship between the plaintext and ciphertext to gain information about the key. He can repeat this analysis on a number of images. The attack uses the fact that key dependent permutations only change the order of coefficients but do not change the values of coefficients and hence if the attacker can create an image which produces distinct values for coefficients in a subband, he can find the permutation applied to that subband by capturing the coefficients in his decoder and observing their order.

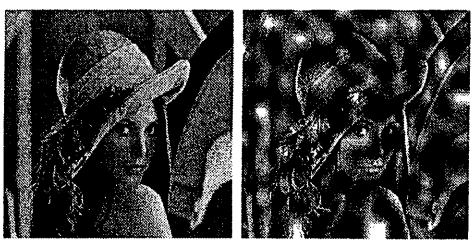

Figure 1: The original image (left) and the recovered image without inverse-permutations when the image is encoded with subband 0 permuted (right).

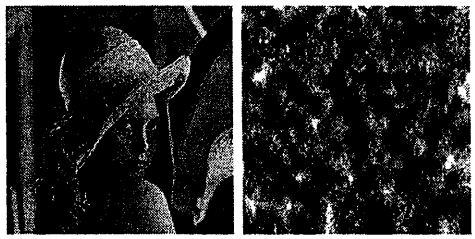

Figure 2: The recovered image without inverse-permutations when the image is encoded with subband 15 permuted (left) and the recovered image without inverse-permutations when the image is encoded with subbands 0 to 15 permuted (right).

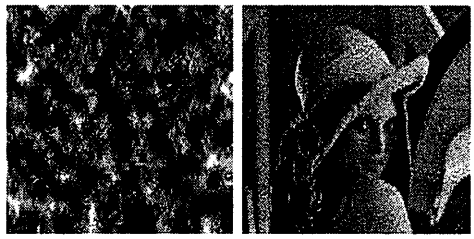

Figure 3: The recovered image without inverse-permutations when the image is encoded with subbands 0 to 7 permuted (left) and the recovered image without inverse-permutations when the image is encoded with subbands 8 to 15 permuted (right).

The assumptions are:

i) The algorithms of transform, quantization, entropy coding and the permutation are known but the key is secret. *ii*) The attacker can obtain ciphertext (compressed images) corresponding to any chosen plaintext (original images). *iii*) The attacker can create an image with coefficients of his choice.

The attack steps will be as follows.

1 The attacker generates an image which has distinct coefficients in one or more subbands.

2 He/she gives the image to the target encoder.

3 The encoder transforms, permutes, quantizes and encodes the image.

4 The attacker obtains the encoded image and decodes the compressed image using his/her decoder.

5 The attacker captures the decoded values between the de-quantization and the inverse-transform and compares the values with his/her chosen values and finds the permutation.

A single attack image in general may recover only part of the permutation.

For the attack to be successful care must be taken against the following types of errors.

i) The pixel values that are obtained from the coefficients must be in the valid range (typically 0 to 255). *ii*) The error in converting real-valued coefficients to integer-valued pixels should not destroy the distinctness of coefficients calculated in the decoder. *iii*) The quantization process in the target encoder should not destroy the distinctness of the coefficients.

In [Dav97], calculation of the transform and inverse-transform has considerable precision and so many distinct coefficients can be chosen. The main restriction is due to the quantization precision. If each of the coefficients in subband b is represented by $q^{(b)}$ bits there can be at most $2^{q^{(b)}}$ distinct coefficients and $2^{q^{(b)}}$ distinct values can reveal the permutation of $2^{q^{(b)}}$ coefficients in a single attack attempt.

This means that coarser subbands are more vulnerable to the attack because, *i*) the quantization precision for the coarser subbands is higher than the detailed ones, and so the number of distinct coefficients that can be used in a single attack will be larger and ii) less number of attack attempts are required for the coarser subbands since coarser subbands have smaller number of coefficients compared with the higher subbands.

5 Enhancing Security

As noted above the chosen plaintext attack is mainly effective for the lowest subband which largely contributes to PSNR in general. To improve security of the scheme we can use a traditional block cipher algorithm such as DES [Ins81] to encrypt coefficients in the lowest subband similar to encrypting DC coefficients of a DCT transformed image in [Tan96]. Using DES ensures that the lowest subband is highly secure, and using permutations for the higher subbands ensures that the detail information are hidden too.

It is interesting to note that the cost of block encryption for wavelet based systems is lower than DCT based systems. The following example clarifies this point.

We compare the wavelet transform with DCT applied to MPEG-1. In MPEG-1, a color image of $m \times n$ pixels is broken into $\frac{m \times n}{16 \times 16}$ macro-blocks, where each of macro-blocks is decomposed into 4 luminance and 2 chrominance 8×8 blocks. This means that the image is represented by $\frac{6mn}{256}$ 8×8 blocks. Each of the 8×8 blocks is DCT-transformed into one 8 bit DC coefficient and 63 AC coefficients. In total, $\frac{48mn}{256} = 0.1875mn$ bits must be encrypted.

In wavelet transform with five 2-D filter banks, we assume the 2 chrominance components have half the size of the luminance component (as in the DCT case). In this case each of the color components will be independently transformed which results in $\frac{mn}{4^5}$ and $\frac{mn}{4 \times 4^5}$ coefficients in the coarsest luminance and chrominance subbands, respectively. So overall, the transform of the 3 color components produces $\frac{5mn}{4^6}$ coefficients. If the quantization precision for the coefficients is 8 bits, this results in $\frac{40mn}{4^6} \approx 0.0098mn$ bits which is approximately 1/20 of the DCT case.

It is worth noting that the above comparison assumes that the information in DC coefficients of 8×8 DCT is almost the same as that of the coarsest subband of the above wavelet transform. To find the exact amount of information in the two cases (DC coefficients of DCT and the lowest subband of wavelet), a more detailed analysis is required.

6 Conclusion

We have shown that permuting one or a small number of subbands in a wavelet based compression system can add security without having a large effect on the compression ratio. By increasing the number of subbands with permuted coefficients the security can be increased and so the system provides variable levels of security. We showed that despite reasonable perceptual masking of the information and the very large size of the key space, the system is vulnerable to a chosen plaintext attack in which a specially constructed image is encoded and the output of the decoder is analyzed. The attack is particularly effective against the lowest subband. We proposed an extension of the system that provides protection against this attack.

References

[ABMD92] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies. Image coding using wavelet transform. *IEEE transactions on image processing*, 1:205–220, Apr 1992.

[Dav97] Geoff Davis. *Baseline wavelet transform coder construction kit.* http://www.cs.dartmouth.edu/~gdavis/wavelet/wavelet.html, 1997.

[Ins81] Amer. Nat. Stand. Inst. *ANSI X.3.92 American National Standard for Data Encryption Algorithm.* 1981.

[SSR99] Sang Uk Shin, Kyeong Seop Sim, and Kyung Hyune Rhee. A secrecy scheme for MPEG video data using the joing of compression and encryption. *ISW'99*, pages 191–201, 1999.

[Tan96] Lei Tang. Methods for encrypting and decrypting MPEG video data efficiently. *In Proceedings of the ACM Multimedia96*, pages 219–229, Nov 1996.